A Blog  
  
Many blog posts that I’ve read over the years tend to quickly narrow their audience in the opening paragraphs. While this has its benefits, I’d like to try and find a balance. I want to keep my posts informative for the technically inclined while still giving new comers a chance to learn.   
Feel free to email me with any topics that you’d like to be discussed. I’m keen to answer any questions, and hear all thoughts/opinions.   
  
Hosting Your Personal Website at Home

3rd August 017

Contents:   
1. Intro - Bought domain name  
2. The Details – DMZ exposing pi to internet

3. Conclusion

Intro

I’ve seen plenty of online portfolios. Especially from undergraduates studying Computer Science. To create my own, I embarked on a slightly more unconventional approach (in 2017 that is).   
  
The Raspberry Pi, an affordable, credit card sized computer was the instrument of choice. I’d bought one during university and it was just gathering dust. In an attempt save it from neglect, I used it to host this personal website.

Although the project was small and took less than a day in total (spent a few hours here and there when I found the time). It was an educational one.

Information consumed at university is often not given appropriate context.   
Depending on the module taken, you may find that much of the examinable material is theoretical. While this is beneficial for many reasons, it tends to commonly raises barriers when one tries to apply their newfound knowledge in a practical setting.

The main goal of this side project was to solidify familiar concepts by building something useful.

Equipment:  
- MacBook Pro 2.4GHz Intel Core i5 16GB RAM - macOS Sierra  
- Raspberry Pi Model B 3

- TalkTalk HUAWEI HG633  
- 32GB MicroSD Card/Reader

- Micro USB cable   
- Ethernet Cable

The Details

In the interest of brevity, the focus of this post will be mainly on the lesser known concepts that you’re likely to encounter. However, I’ll still provide enough detail so that you could try the same at home. It’s also worth mentioning, my intention is not to explain in depth how these technologies work, but to instead give the reader a working knowledge to allow them to understand why and how things are done this way.   
  
1. Purchase a Domain Name  
Evidently, I bought the domain ‘blainemalone.com’. It cost £14.89 from ‘godaddy.com’. I’d advise that you do this at the end of the project if you are unsure that you’ll see it through (in the meantime use the routers IP address to test if website is live). Also, there are other domain registrars. Price around to find the best offers for you. With this said, I did appreciate GoDaddy’s intuitive DNS management dashboard and troubleshooting forums.   
  
2. Set up Raspberry Pi  
Insert your Micro SD card into the reader and then into your laptop/desktop computer. Now we want to download the Raspbian OS from here <https://www.raspberrypi.org/downloads/raspbian/> (select minimal image, only using Pi as a webserver. As a result, we don’t really require a GUI).   
Once downloaded, I used Etcher to burn the .img file to the SD card https://etcher.io/. For more information on how to do this, see <https://www.raspberrypi.org/documentation/installation/installing-images/README.md>.

After you completed this successfully, insert the SD card into the Raspberry Pi, connect the Micro USB for power and attach the Ethernet cable to both the Pi and the router.   
  
  
What you can expect to encounter:  
- Purchasing a domain name  
- Managing DNS configuration  
- Configuring your home network router \*\*  
- SFTP & SSH  
- NAT, PAT  
- DDNS  
- Containerization

- Web servers  
- JavaScript  
-   
  
  
   
\*\* Note: Please be careful when configuring your router at home. Make sure you understand the implications of what you are doing, otherwise you could leave your home network vulnerable. I’ve highlighted some of the obvious vulnerabilities. Please remember, even in the best case, your setup will only ever be as secure as the least secure portion of whatever is accessible over the Internet.